МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | |  | | Руководство программиста  на лабораторную работу №8  по дисциплине «Алгоритмы и структуры данных»  Тема «Разработка программы "Тарифы"» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-22  Линкевич А.К.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**Руководство программиста.**

**Руководство программиста на разработку программы "Тарифы".**

**1. Обзор программы**

Программа представляет собой графический интерфейс (GUI) для управления списком тарифов. Она позволяет:

* Загружать данные о тарифах из файла "tariffs.txt".
* Просматривать список тарифов.
* Выполнять сегментацию(разделение на определенные группы) тарифов по заказчикам и типам услуг.
* Отображать результаты сегментации(разделения на группы) в виде диаграммы.

**2. Структура программы**

Программа состоит из двух классов:

1. Tariff: Представляет собой объект тариф. Имеет следующие атрибуты:

* Tariff\_name: Название тарифа.
* customer: Имя заказчика, использующего тариф.
* service\_type: Название услуги, входящей в тариф.

1. TariffManager: Управляет списком тарифов. Имеет следующие методы:

* init(): Инициализирует менеджер, загружает данные из файла и создает GUI.
* load\_data(): Загружает данные о тарифах из файла.
* segment\_by\_customer(): Выполняет сегментацию(разделение на определенную группу) по заказчикам.
* segment\_by\_service\_type(): Выполняет сегментацию(разделение на определенную группу) по типам услуг.
* create\_gui(): Создает GUI программы.
* show\_customer\_data(): Отображает результаты сегментации по заказчикам.
* show\_service\_data(): Отображает результаты сегментации по услугам.
* visualize\_data(data, title): Отображает данные в виде диаграммы.

**3. Алгоритм работы программы**

1. Инициализация:

* При запуске программы создается экземпляр класса TariffManager.
* TariffManager загружает данные из файла "tariffs.txt".
* TariffManager создает GUI программы.

2. Взаимодействие пользователя:

* Пользователь может нажать на кнопки "Сегментация по заказчикам", "Сегментация по типам услуг".

3. Обработка данных:

* При нажатии на кнопки "Сегментация по заказчикам" или "Сегментация по типам услуг" вызывается соответствующий метод TariffManager ( segment\_by\_customer() или segment\_by\_service\_type()).
* Методы segment\_by\_customer() и segment\_by\_service\_type() возвращают словарь с количеством тарифов для каждого заказчика или типа услуг.
* Результаты сегментации(разделения на группы) используются для создания диаграммы.

4. Отображение результатов:

* Результаты сегментации(разделения на группы) отображаются в виде диаграммы.
* **4. Описание кода**

675.py:

class Tariff:

def init(self, tariff\_name, customer, service\_type):

self.tariff\_name = tariff\_name

self.customer = customer

self.service\_type = service\_type

def str(self):

return f"{self. tariff\_name } - Customer: {self. customer }, Service: {self. service\_type}"

1. Класс Tariff:

* Метод init(self, tariff\_name, customer, service\_type): Инициализирует объект Tariff с заданными параметрами.
* Метод str(self): Возвращает строковое представление объекта Tariff.

675.py:

import tkinter as tk

from tkinter import ttk

import matplotlib.pyplot as plt

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

# ... (код класса TariffManager)

1. Класс TariffManager:

* Метод init(self):
* Инициализирует список tariffs.
* Вызывает load\_data().
* Вызывает create\_gui().
* Вызывает segment\_by\_customer().
* Вызывает segment\_by\_service\_type().
* Вызывает visualize\_data().
* Вызывает show\_customer\_data().
* Вызывает show\_service\_data().
* Метод load\_data(self):
* Загружает данные о тарифах из файла "tariffs.txt".
* Обрабатывает ошибку, если файл не найден.
* Метод segment\_by\_customer(self):
* Выполняет сегментацию по заказчикам.
* Возвращает словарь с количеством заказчиков для каждого типа услуг.
* Метод segment\_by\_service\_type(self):
* Выполняет сегментацию по типам услуг.
* Возвращает словарь с количеством услуг для каждого заказчика.
* Метод create\_gui(self):
* Создает главное окно root.
* Устанавливает заголовок окна.
* Определяет размеры окна и позицию на экране.
* Создает рамку main\_frame для кнопок и отображения данных.
* Создает кнопки "Сегментация по заказчикам", "Сегментация по типам услуг".
* Создает рамку chart\_frame для диаграммы.
* Метод show\_customer\_data(self):
* Вызывает segment\_by\_customer().
* Отображает результаты сегментации в виде диаграммы.
* Метод show\_service\_data(self):
* Вызывает segment\_by\_service\_type().
* Отображает результаты сегментации в виде диаграммы.

**5. Рекомендации по использованию**

* Программа использует файл "tariffs.txt" для хранения данных. Файл должен быть создан вручную и содержать информацию о тарифах в формате:

(Название тарифа, Имя заказчика, Тип услуги)

* Для сегментации по заказчикам или типам услуг достаточно нажать соответствующую кнопку.

**6. Дополнительные возможности**

* Добавить возможность удаления тарифов.
* Добавить возможность редактирования данных о тарифах.
* Добавить возможность использования других типов диаграмм.

**7. Тестирование**

* Тестирование программы должно включать:
* Проверку загрузки данных из файла.
* Проверку сегментации по заказчикам и типам услуг.
* Проверку корректности отображения результатов сегментации.

**8. Документация**

* Данное руководство программиста является основным документом по программе.
* Дополнительную документацию можно добавить в виде комментариев к коду.
* Важно использовать стандартные соглашения по наименованию переменных и функций.
* Документация должна быть четкой, понятной и доступной для пользователя.

**9. Замечания**

* Программа использует библиотеку tkinter для создания GUI.
* Программа использует библиотеку matplotlib для создания диаграмм.
* Программа использует кодировку cp1251 для чтения и записи данных из файла.

**10. Дополнительные замечания**

* Программа может быть расширена и улучшена.
* Программа может быть использована как основа для создания более сложных приложений.
* Важно следовать принципам модульности, повторного использования кода и ясности написания кода.